# Exponential Enclosure Techniques for the Computation of Guaranteed State Enclosures in VALENCIA-IVP*

Andreas Rauh, Ramona Westphal, Harald Aschemann
Chair of Mechatronics, University of Rostock,
Justus-von-Liebig-Weg 6, D-18059 Rostock, Germany
{Andreas.Rauh,Ramona.Westphal,Harald.Aschemann}@uni-rostock.de

Ekaterina Auer
Faculty of Engineering, INKO, University of
Duisburg-Essen, D-47048 Duisburg, Germany
Auer@inf.uni-due.de

**Abstract**

Verified integration of initial value problems for sets of ordinary differential equations can be performed by numerous approaches. The most important ones are based on either interval or Taylor model arithmetic and enclose with certainty the sets of reachable states at a given point of time. Commonly, such tools are based on a Taylor series expansion of the solution of differential equations in time and sometimes in the initial conditions and uncertain parameters. However, the use of high series expansion orders prevents one from applying these tools in real-time environments. This becomes necessary if model-based predictive control strategies are implemented, exploiting an online evaluation of state equations over a finite time horizon. Therefore, it is desirable to reduce the computational effort as far as possible by finding a trade-off between the simplicity of the descriptions of verified state enclosures on the one hand and the amount of overestimation that is contained in the solutions on the other. In VALENCIA-IVP, such enclosures are defined either by interval boxes computed by a simple iteration scheme on the basis of a non-verified numerical approximate solution or by means of exponential enclosures. In this paper, the technique of exponential state enclosures is extended to a novel iteration scheme based on complex-valued interval arithmetic. This iteration reduces overestimation significantly for oscillatory linear state equations. Simulation results for different practically motivated systems with an extension to nonlinear models conclude this paper.

**Keywords:** Complex interval arithmetic, dynamic systems, initial value problems
**AMS subject classifications:** 65G20, 65G30, 65L05, 65L70

*Submitted: February 10, 2013; Revised: July 26, 2013; Accepted: November 22, 2013.

# 1 Introduction

ValEncIA-IVP is a verified solver which computes guaranteed enclosures of exact solutions of initial value problems (IVPs) for systems of ordinary differential equations (ODEs) [23, 3, 22]. Originally, this solver has been implemented on the basis of a simple iteration scheme that allows us to determine guaranteed state enclosures for IVPs with continuously differentiable right-hand sides. These state enclosures are given by a numerically computed approximate solution (for example by means of a classic, i.e., floating point-based, explicit Euler or Runge-Kutta method) with additive guaranteed error bounds. In [24, 22], this solution procedure was extended by an exponential enclosure approach, allowing us to compute tighter state enclosures for asymptotically stable processes.

To efficiently exploit the exponential enclosure approach, the state equations are first decoupled as far as possible. For that purpose, linear dynamic systems are transformed into their real Jordan normal form. After that, the IVP is solved for the equivalent problem. Finally, guaranteed state enclosures in the original coordinates are determined by a suitable verified backward transformation.

However, this decoupling procedure does not manage to eliminate the wrapping effect in cases in which the original system is nonlinear or in cases in which a linear system has an oscillatory behavior. In the case of nonlinear systems, the transformation is determined for a locally linearized model. The reason for the missing capability to eliminate the wrapping effect results from the fact that the transformed system matrix of the (linearized) model is no longer purely diagonal but has a block diagonal structure. Geometrically, each block corresponds to a rotation (and scaling) of state enclosures between two subsequent time steps, which is the reason for the wrapping effect if the state enclosures are propagated over time.

To eliminate the wrapping effect that originates from this rotation, the real-valued IVP with a block diagonal system matrix can be replaced by a transformation into a complex-valued diagonal form if the linear system model does not have multiple eigenvalues. In this contribution, a novel solution procedure for the computation of state enclosures is presented which operates on complex-valued IVPs in the corresponding normal form. This allows us to determine contracting state enclosures for linear ODE systems with asymptotically stable, conjugate complex eigenvalues of multiplicity one by means of a complex-valued exponential enclosure approach with a suitable backward transformation onto the original problem.

This paper is structured as follows: In Sec. 2, the task of verified simulation of sets of ODEs is reviewed briefly with a short summary of possible solution procedures. In Sec. 3, the basic iteration schemes implemented in ValEncIA-IVP are described. Afterwards, Sec. 4 gives an overview of the theory of the novel solution procedure based on complex-valued interval arithmetic. Possible real-life applications originating from modeling of mechatronic systems and from the field of control engineering are presented as benchmark scenarios in Sec. 5. These scenarios include both single and multiple eigenvalues, parameters with bounded uncertainty, and simple nonlinearities. Conclusions and an outlook on future work are given in Sec. 6.

# 2 Verified Integration of Initial Value Problems with Interval Uncertainties

The goal of the numerical procedures presented in this paper is the computation of guaranteed enclosures of all possible states at a given point of time $t$ that belong to the solution of an IVP for a finite-dimensional set of ODEs. Here, uncertainties in both the initial states and system parameters are taken into account by means of interval variables.

## 2.1 Problem Formulation

As a basic setup, the nonlinear continuous-time state equations

$$\dot{\mathbf{x}}_s(\tau) = \tilde{\mathbf{f}}_s(\mathbf{x}_s(\tau), \mathbf{u}(\tau), \mathbf{p}(\tau), \tau) = \mathbf{f}_s(\mathbf{x}_s(\tau), \mathbf{p}(\tau), \tau) \tag{1}$$

are considered with the given initial states $\mathbf{x}_s(0) = \mathbf{x}_{s0}$. Here, $\mathbf{x}_s(\tau)$ denotes the vector of system states, that can be influenced by means of a control signal $\mathbf{u}(\tau) = \mathbf{u}(\mathbf{x}_s(\tau), \tau)$. In many practical applications, feedback controllers $\mathbf{u}(\mathbf{x}_s(\tau), \tau)$ are designed in such a way that the closed-loop system has a dominating linear behavior. Hence, linear systems of ODEs containing uncertain parameters are the focus of this paper[1]. In contrast to $\mathbf{x}_s(\tau)$, the parameter vector $\mathbf{p}(\tau)$ is only subject to external influences, which are not directly controllable.

For the dynamic system (1), uncertainties are taken into account for the initial conditions and parameters according to

$$[\mathbf{x}_{s0}] := [\underline{\mathbf{x}}_{s0} \; ; \; \overline{\mathbf{x}}_{s0}] \qquad \text{and} \qquad [\mathbf{p}(\tau)] := \left[\underline{\mathbf{p}}(\tau) \; ; \; \overline{\mathbf{p}}(\tau)\right] \; . \tag{2}$$

In (1) and (2), it is furthermore possible (cf. [23]) to take into consideration a dynamical model for all time-varying parameters

$$\dot{\mathbf{p}}(\tau) = \Delta\mathbf{p}(\tau) \; , \tag{3}$$

where their range-bounded variation rates are given by

$$\Delta\mathbf{p}(\tau) \in [\Delta\mathbf{p}(\tau)] := \left[\underline{\Delta\mathbf{p}}(\tau) \; ; \; \overline{\Delta\mathbf{p}}(\tau)\right] \; . \tag{4}$$

Since the focus of this paper is the derivation of a novel simulation procedure for this type of systems, the dynamical system model is abbreviated in the following by the ODEs

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) := \begin{bmatrix} \mathbf{f}_s(\mathbf{x}_s(t), \mathbf{p}(t), t) \\ \Delta\mathbf{p}(t) \\ 1 \end{bmatrix} \tag{5}$$

after definition of the extended state vector

$$\mathbf{x}(t) := \begin{bmatrix} \mathbf{x}_s(t) \\ \mathbf{p}(t) \\ t \end{bmatrix} \; , \quad \mathbf{x}(t) \in \mathbb{R}^n \tag{6}$$

[1] Throughout this paper, vectors are denoted by boldface small letters and matrices by uppercase bold letters to distinguish them from scalar variables. Moreover, intervals are denoted explicitly by square brackets, e.g., $[\mathbf{x}] = \left[[x_1] \quad \ldots \quad [x_n]\right]^T$ represents an interval vector with $[x_i] = [\underline{x}_i \; ; \; \overline{x}_i]$, $\underline{x}_i \le x_i \le \overline{x}_i$, $i = 1, \ldots, n$.

with the corresponding initial conditions

$$\mathbf{x}(0) := \mathbf{x}_0 := \begin{bmatrix} \mathbf{x}_s(0) \\ \mathbf{p}(0) \\ 0 \end{bmatrix} \in [\mathbf{x}(0)] := [\mathbf{x}_0] := \begin{bmatrix} [\mathbf{x}_s(0)] \\ [\mathbf{p}(0)] \\ [0\ ;\ 0] \end{bmatrix} . \tag{7}$$

To be able to apply the verified ODE solver ValEncIA-IVP to this type of systems without any specific modifications, it is assumed that the vector-valued function $\mathbf{f}$ defined in (5) is at least once continuously differentiable.

## 2.2 Overview of Different Solution Procedures

In the last decades, numerous verified solution procedures have been developed to compute guaranteed enclosures to the type of IVPs that has been defined in the previous subsection. Based on Moore's basic research studies about the verified solution of IVPs for ODE systems [17], Lohner has developed the solver AWA [14, 15]. It is based on a Taylor series expansion of the solution to the IVP with respect to time, where guaranteed bounds are determined for the resulting discretization errors by means of a Picard iteration. Basically, the corresponding enclosure technique can be derived by means of Banach's fixed point theorem. The fundamental solution procedures in AWA were improved in the solvers VNODE and VNODE-LP [19, 18, 20]. These solvers do not only provide a QR-preconditioning of the state equations (as in AWA) to reduce the wrapping effect but also use the Interval-Hermite-Obreshkoff method instead of simple Taylor expansions (VNODE, VNODE-LP) to further tighten the guaranteed enclosures. Efficient strategies for a reduction of the wrapping effect are necessary since it may lead to unusably wide interval enclosures of the true solution sets in the worst case.

Moreover, further arithmetics were investigated for the description of solution sets. The most successful alternative approach is implemented in COSY VI [5, 4, 16] and RiOT [7], where Taylor models are used to describe both convex and non-convex solution sets. The idea of using Taylor models is also included in VSPODE [13]: Here, a Taylor series expansion is used to account for the time dependency of the solution, while Taylor models are employed to handle the dependency of the solution on (uncertain) initial conditions and parameters. Further approaches aiming at the computation of reliable bounds for the sets of reachable system states make use of ellipsoidal enclosures or zonotopes [12].

However, the common problem of most of the techniques summarized above is the computational effort that may become quite high even for linear systems if tight state enclosures are desired. This computational burden may prevent the online application in predictive control procedures [26, 27]. To implement such procedures, it is necessary to provide verified ODE solvers which are both fast and accurate. Therefore, approaches which do not have an a priori upper bound for the computational effort — or which do not provide solutions in typically a few milliseconds — cannot be employed for this type of application. For these reasons, techniques for the reduction of overestimation using global optimization techniques, forward-backward evaluation of state equations, other consistency tests, or subdivision procedures are typically not applicable in real-time environments [23].

In the following, extensions of the verified ODE solver ValEncIA-IVP are presented which fulfill the requirements with respect to real-time applicability. These extensions make use of the stability properties of the considered systems as well as of

a special structure of the state equations which typically are guaranteed due to widely used design procedures for feedback controllers.

# 3 VALENCIA-IVP — VALidation of state ENClosures using Interval Arithmetic for IVPs

## 3.1 Basic Iteration Scheme

VALENCIA-IVP uses an a posteriori approach to enclose the solution of the IVP (5) with a continuously differentiable right-hand side $\mathbf{f}$. In the basic version of VALENCIA-IVP, the true solution $\mathbf{x}^*(t)$ is enclosed by a tube $[\mathbf{x}](t)$ that consists of a non-verified approximation $\tilde{\mathbf{x}}(t)$ and an iteratively computed verified error bound $[\mathbf{R}](t)$ according to

$$\mathbf{x}^*(t) \in [\mathbf{x}](t) := \tilde{\mathbf{x}}(t) + [\mathbf{R}](t) \ . \tag{8}$$

For the following derivation of the iteration scheme, consider a point in time $t \in [0\ ;\ T]$ with the corresponding exact solution $\mathbf{x}^*(t) \in [\mathbf{x}]([0\ ;\ T])$. In this notation, $[\mathbf{x}](t)$ is an interval inclusion function of the point-valued exact trajectory $\mathbf{x}^*(t)$ at a given point of time $t$, while $[\mathbf{x}]([0\ ;\ T])$ is an interval box that contains all reachable state values over the complete time interval $[0\ ;\ T] \ni t$. As the derivation principle below shows, the function $[\mathbf{x}](t)$ is an interval extension of $\mathbf{x}(t)$ in the sense that $\mathbf{x}(t) \in [\mathbf{x}](t)$ holds for all $t \in [0\ ;\ T]$. Due to the inclusion monotonicity and due to the explicit time dependency of this enclosure, differentiation with respect to $t$ is admissible in such a way that $\dot{\mathbf{x}}(t) \in \frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{x}](t) \subseteq [\dot{\mathbf{x}}](t)$ holds for all $t \in [0\ ;\ T]$. Further details can be found in Appendix A.

If a starting interval $[\mathbf{x}]^{(0)} := [\mathbf{x}]^{(0)}([0\ ;\ T])$ exists such that $\mathbf{x}^*(t) \in [\mathbf{x}]^{(0)}$ holds[2], the exact solution $\mathbf{x}^*(t)$ and an interval inclusion function $[\mathbf{x}](t)$ can be defined according to

$$\mathbf{x}^*(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}\left(\mathbf{x}^*(s)\right) \mathrm{d}s \in [\mathbf{x}_0] + \int_0^t \mathbf{f}\left([\mathbf{x}]^{(0)}\right) \mathrm{d}s \tag{9}$$

$$= [\mathbf{x}_0] + \mathbf{f}\left([\mathbf{x}]^{(0)}\right) \cdot t =: [\mathbf{x}]^{(1)}(t) \ . \tag{10}$$

If $[\mathbf{x}]^{(1)} := [\mathbf{x}]^{(1)}([0\ ;\ T]) \subseteq [\mathbf{x}]^{(0)}$ holds, a Picard iteration is performed according to

$$\mathbf{x}^*(t) \in [\mathbf{x}]^{(\kappa+1)}(t) := [\mathbf{x}_0] + \mathbf{f}\left([\mathbf{x}]^{(\kappa)}\right) \cdot t \ , \quad \kappa \geq 0 \ , \tag{11}$$

where $[\mathbf{x}]^{(\kappa)} := [\mathbf{x}]^{(\kappa)}([0\ ;\ T])$ represents a guaranteed enclosure for $[\mathbf{x}](t)$.

Using the definition (11) of the verified state enclosure, it is desired to compute the interval bounds

$$\mathbf{x}^*(t) \in \tilde{\mathbf{x}}(t) + [\mathbf{R}]^{(\kappa+1)}(t) \quad \text{with} \quad [\mathbf{R}]^{(\kappa)}(0) = [\mathbf{R}]^{(\kappa+1)}(0) = [\mathbf{x}_0] - \tilde{\mathbf{x}}(0) \tag{12}$$

in an iterative scheme such that they correspond to the original definition (8). For that purpose, the term $[\dot{\mathbf{R}}]^{(\kappa+1)}(t)$ is defined as an interval inclusion function of the derivative of $[\mathbf{R}]^{(\kappa+1)}(t)$ according to

$$[\dot{\mathbf{R}}]^{(\kappa+1)}(t) := -\dot{\tilde{\mathbf{x}}}(t) + \mathbf{f}\left([\mathbf{x}]^{(\kappa)}\right) \ , \tag{13}$$

[2]Note that the subscript index 0 always denotes initial values at the point of time $t = 0$, and superscript indices $(\kappa)$, $\kappa \geq 0$, denote the number of an iteration.

where the interval enclosure of the range $\mathbf{f}\left([\mathbf{x}]^{(\kappa)}\right)$ is to be as tight as possible to minimize overestimation. For example, the mean value theorem or monotonicity tests might be used to compute it (cf. [22, 3]).

If the property

$$[\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T]) \subseteq [\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T]) \tag{14}$$

holds, the iteration can continue, leading to the enclosure $[\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T])$ that contains the derivatives of the error term for the complete time interval $[0\ ;\ T] \ni t$. This enclosure is then integrated with respect to time to calculate the error bound itself:

$$\begin{aligned}[\mathbf{R}]^{(\kappa+1)}(t) &\subseteq [\mathbf{R}]^{(\kappa+1)}(0) + \int_0^t [\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T])\,\mathrm{d}s \\ &= [\mathbf{R}]^{(\kappa+1)}(0) + [\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T]) \cdot t\ ,\end{aligned} \tag{15}$$

that is,

$$\begin{aligned}[\mathbf{R}]^{(\kappa+1)}(t) &\subseteq [\mathbf{R}]^{(\kappa+1)}(0) + [\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T]) \cdot [0\ ;\ T] \\ &=: [\mathbf{R}]^{(\kappa+1)}([0\ ;\ T])\ .\end{aligned} \tag{16}$$

By combining (8) and (16), the verified enclosure $[\mathbf{x}]^{(\kappa+1)}$ given in (8) and (12) is obtained over the interval $[0\ ;\ T] \ni t$, assuming that the approximation $\tilde{\mathbf{x}}(t)$ can be determined using a method from usual floating-point numerics. Note that the last equality in (16) corresponds to a verified integration of the time-invariant interval bounds of $[\dot{\mathbf{R}}]^{(\kappa+1)}([0\ ;\ T])$, see also Appendix A.

The enclosure of the set of all reachable states at $t = T$ then is given by a re-evaluation of (15) for this point of time, where $[\dot{\mathbf{R}}]([0\ ;\ T])$ has been determined for the complete interval $[0\ ;\ T]$. Extensions of this basic approach to systems with non-smooth right-hand sides can be found in [2].

## 3.2 ValEncIA-IVP — Exponential State Enclosures

To prevent the growth of the interval diameters characterizing the verified solutions determined by ValEncIA-IVP especially for asymptotically stable dynamic systems with a minimum computational effort, the exponential state enclosures

$$\mathbf{x}^*(t) \in [\mathbf{x}_e](t) := \exp([\mathbf{\Lambda}] \cdot t) \cdot [\mathbf{x}_e](0) \quad \text{with} \quad 0 \notin [x_{e,i}](0)\ ,\ \ [\mathbf{x}_e](0) = [\mathbf{x}_0] \tag{17}$$

and

$$[\mathbf{\Lambda}] := \operatorname{diag}\{[\lambda_i]\}\ ,\ i = 1, \ldots, n \tag{18}$$

can be used in an extended version of this solver. In this case, an iteration scheme has to be derived for the coefficients $\lambda_i$ that are used in (17) and (18). In (17), the interval matrix exponential is only required for diagonal matrices. It can be defined according to

$$\exp([\mathbf{\Lambda}] \cdot t) := \operatorname{diag}\{\exp([\lambda_1] \cdot t), \ldots, \exp([\lambda_n] \cdot t)\}\ . \tag{19}$$

As shown in Subsection 3.1, a Picard iteration

$$\mathbf{x}^*(t) \in [\mathbf{x}_e]^{(\kappa+1)}(t) := [\mathbf{x}_0] + \int_0^t \mathbf{f}\left([\mathbf{x}_e]^{(\kappa)}(s)\right)\mathrm{d}s \tag{20}$$

is used to determine guaranteed enclosures of the exact solution $\mathbf{x}^*(t)$. If this iteration (20) is evaluated for the exponential state enclosures (17) and (18), the explicitly time-dependent expression

$$\begin{aligned} \mathbf{x}^*(t) &\in \exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot t\right) \cdot [\mathbf{x}_e]\,(0) = [\mathbf{x}_e]^{(\kappa+1)}\,(t) \\ &=: [\mathbf{x}_0] + \int_0^t \mathbf{f}\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot s\right) \cdot [\mathbf{x}_e]\,(0)\right) \mathrm{d}s \end{aligned} \tag{21}$$

is obtained.

Its differentiation with respect to time (which is the upper bound of the above-stated integration (21)) yields

$$\begin{aligned} \dot{\mathbf{x}}^*(t) &\in \mathrm{diag}\left\{[\lambda_i]^{(\kappa+1)}\right\} \cdot \exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot t\right) \cdot [\mathbf{x}_e]\,(0) \\ &:= \mathbf{f}\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot t\right) \cdot [\mathbf{x}_e]\,(0)\right) \quad . \end{aligned} \tag{22}$$

Now, the evaluation of (22) is replaced by an evaluation for the complete time interval $t \in [0\ ;\ T]$ according to

$$\begin{aligned} \dot{\mathbf{x}}^*\left([0\ ;\ T]\right) &\in \mathrm{diag}\left\{[\lambda_i]^{(\kappa+1)}\right\} \cdot \exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0) \\ &\subseteq \mathbf{f}\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right) \quad . \end{aligned} \tag{23}$$

In the case that (21), (22), and (23) describe a converging iteration process with $[\mathbf{x}_e]^{(\kappa+1)} \subseteq [\mathbf{x}_e]^{(\kappa)}$, the expression

$$\exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot t\right) \cdot [\mathbf{x}_e]\,(0) \subseteq \exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot t\right) \cdot [\mathbf{x}_e]\,(0) \tag{24}$$

holds. According to Appendix B, the inclusion monotonicity of the exponential functions implies

$$\exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot t\right) \subseteq \exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot t\right) \tag{25}$$

and thus also

$$\exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot [0\ ;\ T]\right) \subseteq \exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \quad . \tag{26}$$

This is equivalent to the relations

$$[\lambda_i]^{(\kappa+1)} \subseteq [\lambda_i]^{(\kappa)} \qquad \text{and} \qquad [\mathbf{\Lambda}]^{(\kappa+1)} \subseteq [\mathbf{\Lambda}]^{(\kappa)} \quad . \tag{27}$$

For that reason, a conservative over-approximation of the interval $[\lambda_i]^{(\kappa+1)}$ by $[\tilde{\lambda}_i]^{(\kappa+1)}$ can be determined for the complete time interval $t \in [0\ ;\ T]$ on the left-hand side of (23) according to

$$\begin{aligned} \mathrm{diag}\left\{[\tilde{\lambda}_i]^{(\kappa+1)}\right\} \cdot \underbrace{\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)}_{[\mathbf{x}_e]^{(\kappa)}([0\ ;\ T])} \\ := \mathbf{f}\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right) \quad . \end{aligned} \tag{28}$$

Since $[\lambda_i]^{(\kappa+1)} \subseteq [\tilde{\lambda}_i]^{(\kappa+1)}$ holds in the case of convergence of the Picard iteration (i.e., $[\lambda_i]^{(\kappa+1)} \subseteq [\lambda_i]^{(\kappa)}$, see Appendix B), the iteration formula for $[\mathbf{\Lambda}_i]$ can be defined as

$$[\lambda_i]^{(\kappa+1)} := \frac{f_i\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right)}{\exp\left([\lambda_i]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)}\ ,\ i = 1, \ldots, n\ . \tag{29}$$

The solution of all reachable states at $t = T$ then is given by

$$\mathbf{x}^*(T) \in [\mathbf{x}_e]\,(T) := \exp\left([\mathbf{\Lambda}] \cdot T\right) \cdot [\mathbf{x}_e]\,(0)\ \ , \tag{30}$$

where $[\mathbf{\Lambda}]$ is the final result of the iteration (29).

Note that this iteration formula and its derivation are only admissible if $0 \notin [x_{e,i}]\,([0\ ;\ T])$ holds for all $i = 1, \ldots, n$ over the complete time interval $t \in [0\ ;\ T]$. This prerequisite can be checked easily by first applying the basic iteration summarized in Subsection 3.1.

For linear state equations

$$f_i\,(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} \cdot x_j\ \ , \tag{31}$$

the iteration formula (29) can be rewritten as

$$[\lambda_i]^{(\kappa+1)} := \sum_{j=1, i\neq j}^{n} \left\{ a_{ij} \cdot \exp\left(\left([\lambda_j]^{(\kappa)} - [\lambda_i]^{(\kappa)}\right) \cdot [0\ ;\ T]\right) \cdot \frac{[x_{e,j}]\,(0)}{[x_{e,i}]\,(0)} \right\} + a_{ii} \tag{32}$$

to reduce overestimation for the interval-based evaluation for long time intervals $[0\ ;\ T]$ with $a_{ij} \in [a_{ij}]$. A more detailed analysis of the applicability of (29) and (32) and an extension by preconditioning are given in the following. Contracting state enclosures can be expected for dynamic systems (5) and (6) in which $\mathbf{f}$ is linear according to (31) and additionally describes asymptotically stable dynamics.

## 3.3 Preconditioning of the State Equations

The exponential enclosure approach is most efficient if the state equations are dominated by asymptotically stable, linear dynamics and if they are decoupled as far as possible. Moreover, $\mathbf{f}\,(\mathbf{0}) = \mathbf{0}$ should be guaranteed by a suitable choice of the coordinate system. For that reason, linear dynamic systems (31) are transformed into their real Jordan normal form according to the procedure described in [8].

For systems with $n$ real, pairwise different eigenvalues, the Jordan normal form is given by

$$\dot{\mathbf{z}}(t) = \mathbf{\Sigma} \cdot \mathbf{z}(t) \quad \text{with} \quad \mathbf{\Sigma} = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \lambda_n \end{bmatrix} \quad \text{and} \quad \mathbf{z}(0) \in [\mathbf{z}(0)]\ . \tag{33}$$

For systems with real eigenvalues, where $\lambda_i$ has the multiplicity $\delta_i > 1$ and all

other eigenvalues are pairwise different, the Jordan normal form is given by

$$\begin{aligned} \dot{\mathbf{z}}(t) &= \mathbf{\Sigma} \cdot \mathbf{z}(t) \quad \text{with} \quad \mathbf{\Sigma} = \text{blkdiag}\{\lambda_1, \lambda_2 \ldots, \Sigma_i, \ldots \lambda_n\} \ , \\ \mathbf{\Sigma}_i &= \begin{bmatrix} \lambda_i & 1 & \ldots & 0 \\ 0 & \lambda_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \ldots & 0 & \lambda_i \end{bmatrix} \in \mathbb{R}^{\delta_i \times \delta_i} \ , \ \text{and} \quad \mathbf{z}(0) \in [\mathbf{z}(0)] \ . \end{aligned} \tag{34}$$

According to [8, 9, 10], further block diagonal entries are added to $\mathbf{\Sigma}$ which have the same structure as $\mathbf{\Sigma}_i$, if a linear dynamic system has more than one multiple real eigenvalue.

For nonlinear systems as well as for uncertain linear systems, the transformation is performed using the matrix of the eigenvectors of the system's Jacobian that is evaluated for the interval midpoints of all uncertain variables.

It is well known that the exact solution to the case (33) is given by

$$z_i^*(t) = e^{\lambda_i t} \cdot z_i(0) \ , \quad i = 1, \ldots, n \ . \tag{35}$$

Even if the parameters $\lambda_i$ and the initial conditions $z_i(0)$ are uncertain with $\lambda_i \in [\lambda_i]$ ($[\lambda_i] \cap [\lambda_j] = \emptyset$, $i, j = 1, \ldots, n$, $i \neq j$) and $z_i(0) \in [z_i(0)]$ ($0 \notin [z_i(0)]$), $z_i(t) \neq 0$ holds for any $t > 0$. Thus, the exponential enclosure technique of VALENCIA-IVP is always applicable in this case.

For multiple real eigenvalues $\lambda_i$ of multiplicity $\delta_i$, corresponding to case (34), the solutions are given by

$$z_{i+j}^*(t) = e^{\lambda_i t} \cdot \left( \sum_{\zeta=j}^{\delta_i - 1} \frac{t^{\zeta - j}}{(\zeta - j)!} \cdot z_{i+\zeta}(0) \right) \ , \quad j = 0, \ldots, \delta_i - 1 \ . \tag{36}$$

Here, the exponential enclosure technique of VALENCIA-IVP is applicable if

$$0 \notin \left\{ \chi_j \middle| \chi_j = \sum_{\zeta=j}^{\delta_i - 1} \frac{t^{\zeta - j}}{(\zeta - j)!} \cdot z_{i+\zeta}(0), \ z_{i+\zeta}(0) \in [z_{i+\zeta}(0)] \, , \ t \in [0 \, ; \, T] \right\} \tag{37}$$

holds for any $j = 0, \ldots, \delta_i - 1$.

If this transformation into real Jordan normal form is applied to systems with conjugate complex eigenvalues $\lambda_{i,i+1}^* = \sigma_i \pm j\omega_i$ of multiplicity $\delta_i = 1$, the transformed linear system is characterized by a block diagonal structure of its system matrix according to

$$\mathbf{\Sigma} = \text{blkdiag}\{\ldots, \bar{\mathbf{\Sigma}}_i, \ldots\} \ , \quad \bar{\mathbf{\Sigma}}_i = \begin{bmatrix} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{bmatrix} \ . \tag{38}$$

Then, the above-mentioned advantageous property of the solutions (35) not containing the value zero in any component $z_i(t)$ is no longer valid since the solutions contain the terms $\sin(\omega_i t)$ and $\cos(\omega_i t)$, which change their sign periodically according to the system's eigenfrequency $\omega_i$. In such cases, a transformation of the state equations into their complex Jordan normal form according to the procedure in Sec. 4 helps to resolve this problem.

# 4 Exponential State Enclosures Based on Complex-Valued Interval Arithmetic

In analogy to the decoupling of linear state equations by means of a transformation into real Jordan normal form, it is possible to replace the block diagonal entries (38) for systems with pairwise different conjugate complex eigenvalues corresponding to the structure of the system model (31) by the equivalent complex Jordan normal form

$$\mathbf{\Sigma} = \text{blkdiag}\{\ldots, \mathbf{\Sigma}_i, \ldots\} \ , \quad \mathbf{\Sigma}_i = \begin{bmatrix} \sigma_i + j\omega_i & 0 \\ 0 & \sigma_i - j\omega_i \end{bmatrix} \ . \tag{39}$$

For conjugate complex eigenvalue pairs of multiplicity $\delta_i > 1$, the complex-valued Jordan normal form results in

$$\mathbf{\Sigma} = \text{blkdiag}\{\ldots, \mathbf{\Sigma}_i^+, \mathbf{\Sigma}_i^-, \ldots\} \tag{40}$$

with

$$\mathbf{\Sigma}_i^+ = \begin{bmatrix} \sigma_i + j\omega_i & 1 & \ldots & 0 \\ 0 & \sigma_i + j\omega_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \ldots & 0 & \sigma_i + j\omega_i \end{bmatrix} \in \mathbb{C}^{\delta_i \times \delta_i} \tag{41}$$

and

$$\mathbf{\Sigma}_i^- = \begin{bmatrix} \sigma_i - j\omega_i & 1 & \ldots & 0 \\ 0 & \sigma_i - j\omega_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \ldots & 0 & \sigma_i - j\omega_i \end{bmatrix} \in \mathbb{C}^{\delta_i \times \delta_i} \tag{42}$$

for each eigenvalue pair $\sigma_i \pm j\omega_i$ with $\delta_i > 1$.

Now, the corresponding complex-valued initial value problem $\dot{\mathbf{z}}(t) = \mathbf{\Sigma} \cdot \mathbf{z}(t)$ with $\mathbf{z}(0) \in \mathbb{C}^n$, $\mathbf{z}(0) \in [\mathbf{z}(0)]$ is solved by an application of either the basic iteration scheme of ValEncIA-IVP or by its exponential approach. In both cases, the iteration formulas (13), (16), and (29) are evaluated in complex interval arithmetic [21]. In this paper, the midpoint-radius implementation provided by IntLab [29] is used for this purpose. After the evaluation of these iteration formulas, the solution enclosures are transformed back into the original coordinates by left multiplying the solutions by an interval-based, guaranteed inverse of the matrix that has been used for the transformation of the state equations into the (complex) Jordan normal form and by taking the real part of the solution of the before-mentioned product. Note that the techniques for reduction of overestimation (mentioned in Subsection 3.1), which are based on monotonicity tests, are not yet generalized for the complex case.

Now, the applicability of the complex iteration scheme is analyzed as in Subsection 3.3. For conjugate complex eigenvalue pairs of multiplicity $\delta_i = 1$, the solution of the transformed state equations (39) is given by

$$\begin{aligned} z_i(t) &= e^{(\sigma_i + j\omega_i)\cdot t} \cdot z_i(0) \\ z_{i+1}(t) &= e^{(\sigma_i - j\omega_i)\cdot t} \cdot z_{i+1}(0) \ . \end{aligned} \tag{43}$$

The computation of the squared absolute value of $z_i(t)$ yields

$$|z_i(t)|^2 = \left(e^{(\sigma_i + j\omega_i)\cdot t} \cdot e^{(\sigma_i - j\omega_i)\cdot t}\right) \cdot |z_i(0)|^2 = e^{2\sigma_i t} \cdot |z_i(0)|^2 \ . \tag{44}$$

Hence, the complex iteration scheme based on (29) is applicable for any $t \geq 0$ if $0 \notin |[z_i(0)]|$ holds for the evaluation of $|z_i(0)|$ with the initial state intervals given at $t = 0$. Taking into account the relation $|z_i(0)|^2 = \Re\{z_i(0)\}^2 + \Im\{z_i(0)\}^2$, this condition can be replaced by the precondition that

$$0 \in \Re\{[z_i(0)]\} \qquad \text{and} \qquad 0 \in \Im\{[z_i(0)]\} \tag{45}$$

must not hold simultaneously. This requirement typically is fulfilled in the case (38) and (39), where the exponential enclosure techniques failed for a real-valued system model.

This analysis can be carried out analogously for conjugate complex eigenvalue pairs (40) of multiplicity $\delta_i > 1$. Then, the corresponding solutions $z_{i+j}(t)$, $j = 0, \ldots, \delta_i - 1$, are given by

$$z^*_{i+j}(t) = e^{(\sigma_i + j\omega_i)\cdot t} \cdot \left( \sum_{\zeta=j}^{\delta_i - 1} \frac{t^{\zeta - j}}{(\zeta - j)!} \cdot z_{i+\zeta}(0) \right) \quad , \quad j = 0, \ldots, \delta_i - 1 \ . \tag{46}$$

The square of its absolute value is given by

$$|z^*_{i+j}(t)|^2 = e^{2\sigma_i \cdot t} \cdot \left| \left( \sum_{\zeta=j}^{\delta_i - 1} \frac{t^{\zeta - j}}{(\zeta - j)!} \cdot z_{i+\zeta}(0) \right) \right|^2 = e^{2\sigma_i \cdot t} \cdot |\chi_j|^2 \quad . \tag{47}$$

Hence, the exponential enclosure technique of ValEncIA-IVP is applicable if

$$0 \notin \left\{ \chi_j \middle| \chi_j = \Re\{\chi_j\} + j\Im\{\chi_j\},\ z_{i+\zeta}(0) \in [z_{i+\zeta}(0)]\,,\ t \in [0\ ;\ T] \right\} \tag{48}$$

holds for any $j = 0, \ldots, \delta_i - 1$.

# 5 Application Scenarios

In this section, the practical applicability of the exponential enclosure approach in ValEncIA-IVP is demonstrated for several engineering examples. The resulting state enclosures are compared qualitatively with the results that are obtained with VNODE-LP. In ValEncIA-IVP, a constant integration step size is chosen with $\Delta t = 0.001$, while VNODE-LP uses an automatic step-size control with the series expansion order 20.

## 5.1 Basic Models of Oscillatory Dynamics in Electrical and Mechanical Systems

The first application is a typical linear system model with conjugate complex eigenvalues. It arises in many situations in control of electrical and mechanical systems. In electrical engineering, the following type of state equations resembles an RLC series oscillator (resistor, inductance, capacitance) which may represent an equivalent circuit model of an electric drive. As a mechanical system, the system model

$$\ddot{z}\,(t) + \frac{b_D}{m}\,\dot{z}\,(t) + \frac{c_{F1} + c_{F2}}{m}\,z\,(t) = \frac{c_{F2}}{m}\,u\,(t) \tag{49}$$

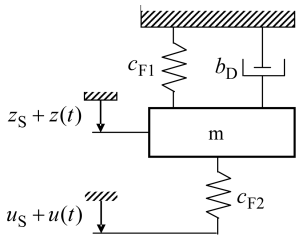


Figure 1: Schematic representation of a quarter vehicle model.

describes a spring-mass-damper system, which is a typical quarter vehicle model (see Fig. 1) and is used widely in the design of strategies for active and passive oscillation damping.

Choosing the parameters $m = 1$, $b_D = 3$, $c_{F1} = 5$, $c_{F2} = 5$ and introducing the state vector $\mathbf{x}'(t) = \begin{bmatrix} z\,(t) \\ \dot{z}\,(t) \end{bmatrix}$, we obtain the state-space representation

$$\dot{\mathbf{x}}'(t) = \begin{bmatrix} 0 & 1 \\ -10 & -6 \end{bmatrix} \mathbf{x}'(t) + \begin{bmatrix} 0 \\ 5 \end{bmatrix} u\,(t) \tag{50}$$

with the eigenvalues $\lambda_{1,2} = -3 \pm 1j = \sigma \pm \omega j$. These state equations are now transformed automatically into their real Jordan normal form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} \sigma_1 & \omega_1 \\ -\omega_1 & \sigma_1 \end{bmatrix}\ , \quad \sigma_1 = -3\ , \quad \omega_1 = 1 \tag{51}$$

with $u(t) = 0$.

## 5.2 Example 1

Using the initial conditions

$$\mathbf{x}(0) \in \begin{bmatrix} [0.9\ ;\ 1.1] \\ [0.9\ ;\ 1.1] \end{bmatrix} \tag{52}$$

for the state equations (51), guaranteed enclosures of $\mathbf{x}(t)$ can only be determined by the exponential approach in VALENCIA-IVP until the point of time at which the value zero is included in the enclosure for $x_2(t)$, see Fig. 2. After that point of time, only the basic iteration scheme is applicable (as long as the value zero is included in any of the state enclosures, unless no interval subdivision is performed).

Transforming the state equations (51) into their complex Jordan normal form leads to

$$\dot{\mathbf{z}}(t) = \tilde{\mathbf{A}}\mathbf{z}(t) \quad \text{with} \quad \tilde{\mathbf{A}} = \begin{bmatrix} \sigma_1 + j\omega_1 & 0 \\ 0 & \sigma_1 - j\omega_1 \end{bmatrix}\ , \quad \sigma_1 = -3\ , \quad \omega_1 = 1 \tag{53}$$

with the initial conditions

$$\mathbf{z}(0) \in \begin{bmatrix} \langle \frac{1}{2}\sqrt{2} \cdot (1 - j)\,, \frac{1}{10}\sqrt{2} \rangle \\ \langle \frac{1}{2}\sqrt{2} \cdot (1 + j)\,, \frac{1}{10}\sqrt{2} \rangle \end{bmatrix} \tag{54}$$

given in their midpoint-radius representation. All computations in the new iteration scheme of VALENCIA-IVP are performed in this section by means of the midpoint-radius arithmetic that is available in INTLAB as a toolbox for MATLAB [29].

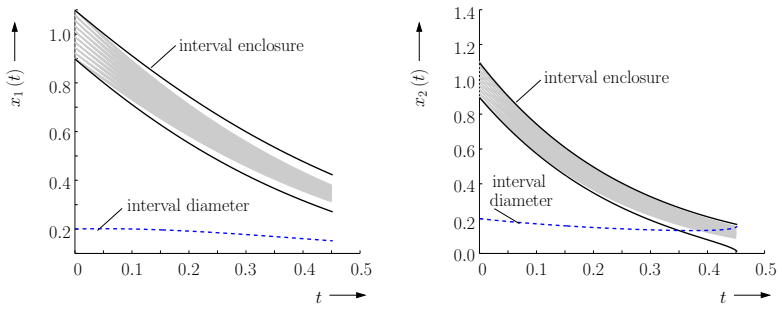


Figure 2: State enclosures computed by the real-valued exponential enclosure approach in VALENCIA-IVP (example 1) in comparison with a grid-based reference solution, for which each of the uncertain initial state intervals has been gridded independently into 10 equally spaced points (gray).

The resulting state enclosures for the complex enclosure approach are shown in Fig. 3. The corresponding results of VNODE-LP are displayed in Fig. 4. The comparison shows that the initial interval widths of the VALENCIA-IVP enclosures are significantly wider than the ones obtained by VNODE-LP.

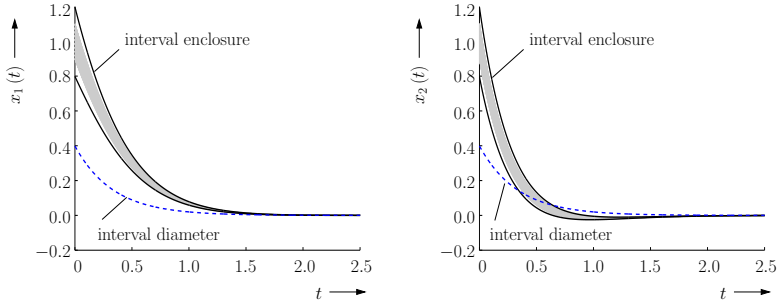


Figure 3: State enclosures computed by the complex-valued exponential enclosure approach in VALENCIA-IVP (example 1).

This is caused by the fact that the axis-parallel interval vector (52) first has to be mapped into the complex plane by the state transformation from $\mathbf{x}$ to $\mathbf{z}$, and later on has to be mapped back. This leads to an additional wrapping of intervals as long as the initial state domains at $t = 0$ are given by intervals and not by discs in midpoint-radius form. However, it is possible to intersect the results of the complex-valued iteration formula (after backward transformation) with the results of the real-valued exponential enclosure approach in a future implementation. The fact that only function evaluations and no time-consuming computations of Taylor coefficients are necessary in the exponential enclosure approach in VALENCIA-IVP improves its usability for real-time applications. Besides the above-mentioned interval enclosures, Figs. 3 and 4 also contain point-valued solutions to the problem (51) and (52), where each of the interval parameters has been gridded into 10 independent, equally spaced points (curves in gray color).

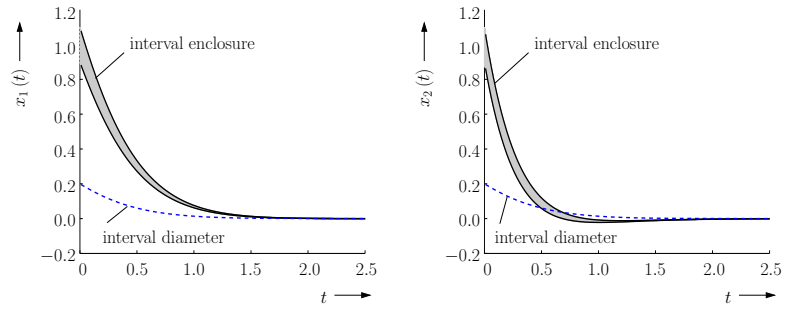


Figure 4: State enclosures computed by VNODE-LP (example 1).

## 5.3 Example 2

To show that the novel extension of ValEncIA-IVP is also applicable to systems with uncertain eigenvalues, the dynamic system model

$$\dot{\mathbf{z}}(t) = \tilde{\mathbf{A}}\mathbf{z}(t) \quad \text{with} \quad \tilde{\mathbf{A}} = \begin{bmatrix} \sigma_1 + j\omega_1 & 0 \\ 0 & \sigma_1 - j\omega_1 \end{bmatrix}, \quad \sigma_1 = -3\,, \quad \omega_1 \in [0.95\,;\,1.05] \tag{55}$$

with the initial conditions

$$\mathbf{z}(0) \in \begin{bmatrix} \langle \frac{1}{2}\sqrt{2} \cdot (1-j)\,, \frac{1}{10}\sqrt{2} \rangle \\ \langle \frac{1}{2}\sqrt{2} \cdot (1+j)\,, \frac{1}{10}\sqrt{2} \rangle \end{bmatrix} \tag{56}$$

in midpoint-radius form is considered.

The resulting state enclosures of ValEncIA-IVP and VNODE-LP are shown in Figs. 5 and 6, including also grid-based solutions in gray color as in to Example 1. For a more detailed discussion, see the end of Sec. 5.2.

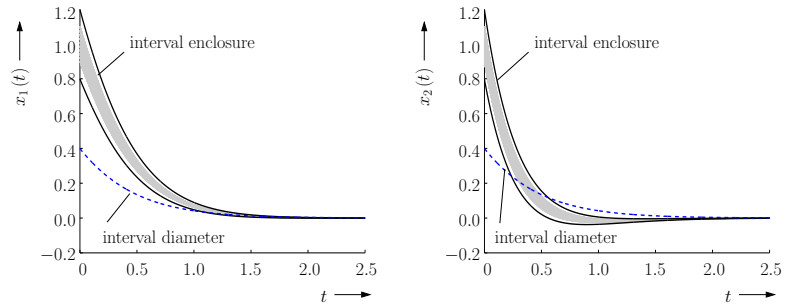


Figure 5: State enclosures computed by the complex-valued exponential enclosure approach in ValEncIA-IVP for a system with uncertain eigenvalues (example 2).

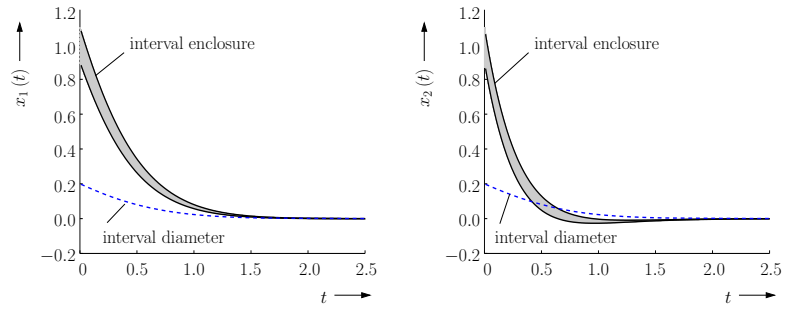


Figure 6: State enclosures computed by VNODE-LP for a system with uncertain eigenvalues (example 2).

## 5.4 Example 3

As a final introductory example, the dynamic system model

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \begin{bmatrix} \sin\left(x_1(t)\right) \\ 0 \end{bmatrix} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} \sigma_1 & \omega_1 \\ -\omega_1 & \sigma_1 \end{bmatrix}, \quad \sigma_1 = -3\,, \quad \omega_1 = 1 \tag{57}$$

with the initial conditions $\mathbf{x}(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ is considered. Here, the transformation matrix into complex Jordan normal form in VALENCIA-IVP is determined for the linear part $\mathbf{A}\mathbf{x}(t)$. Since the linear part has dominant influence on the system dynamics, the novel exponential enclosure approach in VALENCIA-IVP yields results comparable to those from VNODE-LP, see Figs. 7 and 8.

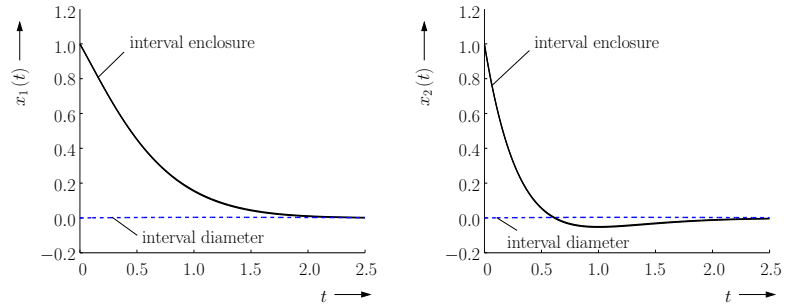


Figure 7: State enclosures computed by the complex-valued exponential enclosure approach in VALENCIA-IVP for a system with nonlinearities (example 3).

## 5.5 Robustness Verification of Linear Closed-Loop Control Procedures for Flexible High Speed Rack Feeders

As a final application scenario, the robustness analysis of a simplified closed-loop control procedure of a high-bay rack feeder system taken from [1] is considered, see Fig. 9.

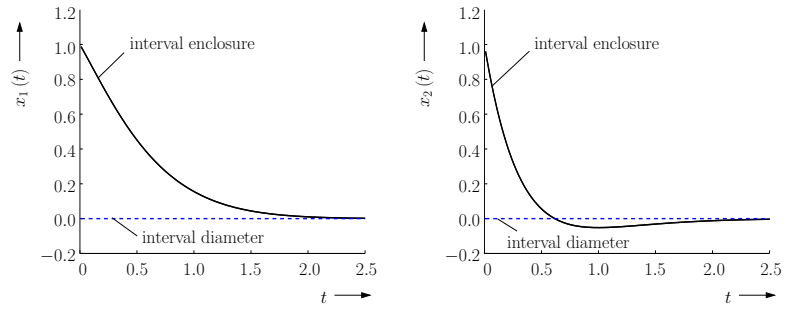


Figure 8: State enclosures computed by VNODE-LP for a system with nonlinearities (example 3).

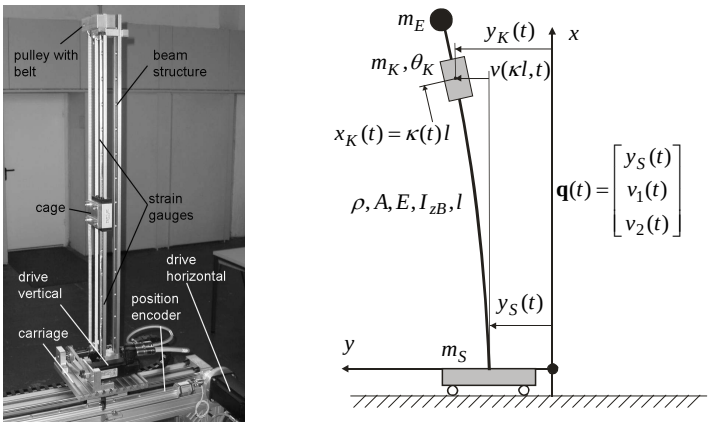


Figure 9: Experimental set-up of the high-speed rack feeder (left) and the corresponding elastic multibody model (right).

During the mathematical description of this rack feeder, which is built up as a test rig at the Chair of Mechatronics at the University of Rostock, an elastic multibody system model was derived to design a feedback control structure. The mathematical description for the rack feeder can be derived as a model with three rigid bodies, consisting of a carriage (mass $m_S$), a cage movable on a vertical double beam structure (mass $m_K$, mass moment of inertia $\theta_K$), and the end mass at the tip of the beam (mass $m_E$), and an elastic Bernoulli beam (density $\rho$, cross sectional area $A$, Youngs modulus $E$, second moment of area $I_{zB}$, and length $l$). The varying vertical position $x_K(t)$ of the cage on the beam is denoted by the dimensionless system parameter

$$\kappa(t) = \frac{x_K(t)}{l} \quad . \tag{58}$$

The elastic degrees of freedom of the beam concerning the bending deflection can be

described by the following Ritz ansatz

$$v(x,t) = \begin{bmatrix} \bar{\bar{v}}_1(x) & \bar{\bar{v}}_2(x) \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} , \tag{59}$$

with

$$\bar{\bar{v}}_1(x) = \frac{3}{2}\left(\frac{x}{l}\right)^2 - \frac{1}{2}\left(\frac{x}{l}\right)^3 \qquad \text{and} \qquad \bar{\bar{v}}_2(x) = \left(\frac{x}{l}\right)^2 , \tag{60}$$

which takes into account the first and the second bending mode. The vector of generalized coordinates results in

$$\mathbf{q}(t) = \begin{bmatrix} y_S(t) & v_1(t) & v_2(t) \end{bmatrix}^T . \tag{61}$$

The nonlinear equations of motion can be derived either by Lagrange's equations or, more efficiently, by the Newton-Euler approach, cf. [30]. For this purpose, position vectors to the corresponding centers of gravity are introduced: the position vector to the carriage $\mathbf{r}_S$, to the cage $\mathbf{r}_K$, to the mass at the beam tip $\mathbf{r}_E$, and to a mass element of the Bernoulli beam $\mathbf{r}_{BE}$ are given by

$$\mathbf{r}_S = \begin{bmatrix} 0 \\ y_S \end{bmatrix}, \mathbf{r}_K = \begin{bmatrix} x_K \\ y_S + v(x_K) \end{bmatrix}, \mathbf{r}_E = \begin{bmatrix} l \\ y_S + v(l) \end{bmatrix}, \mathbf{r}_{BE} = \begin{bmatrix} x_{BE} \\ y_S + v(x_{BE}) \end{bmatrix} . \tag{62}$$

By computing the Jacobians of translation

$$\mathbf{J}_{Ti} = \frac{\partial \mathbf{r}_i}{\partial \mathbf{q}},\ i = \{S, K, E, BE\} \tag{63}$$

for the vectors defined in (62) and the Jacobian of rotation

$$\mathbf{j}_{Rj} = \frac{\partial \varphi_j}{\partial \mathbf{q}},\ j = \{K, E, BE\} \tag{64}$$

for the angles, the nonlinear equations of the motion

$$\tilde{\mathbf{M}}(\mathbf{q})\,\ddot{\mathbf{q}} + \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) = \tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, F_{SM}, F_{SR})$$

are obtained as follows

$$\begin{aligned} \tilde{\mathbf{M}}(\mathbf{q}) = {} & m_S \mathbf{J}_{TS}^T \mathbf{J}_{TS} + m_K \mathbf{J}_{TK}^T \mathbf{J}_{TK} + m_E \mathbf{J}_{TE}^T \mathbf{J}_{TE} + \theta_K \mathbf{j}_{RK} \mathbf{j}_{RK}^T \\ & + \rho \int_0^l \left( A \mathbf{J}_{TBE}^T \mathbf{J}_{TBE} + I_{zB} \mathbf{j}_{RBE} \mathbf{j}_{RBE}^T \right) \mathrm{d}x , \end{aligned} \tag{65}$$

$$\begin{aligned} \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) = {} & m_S \mathbf{J}_{TS}^T \dot{\mathbf{J}}_{TS} \dot{\mathbf{q}} + m_K \mathbf{J}_{TK}^T \dot{\mathbf{J}}_{TK} \dot{\mathbf{q}} + m_E \mathbf{J}_{TE}^T \dot{\mathbf{J}}_{TE} \dot{\mathbf{q}} \\ & + \theta_K \mathbf{j}_{RK} \frac{\mathrm{d}}{\mathrm{d}t} \left( \mathbf{j}_{RK}^T \right) \dot{\mathbf{q}} \\ & + \rho \int_0^l \left( A \mathbf{J}_{TBE}^T \dot{\mathbf{J}}_{TBE} \dot{\mathbf{q}} + I_{zB} \mathbf{j}_{RBE} \frac{\mathrm{d}}{\mathrm{d}t} \left( \mathbf{j}_{RBE}^T \right) \dot{\mathbf{q}} \right) \mathrm{d}x , \end{aligned} \tag{66}$$

$$\tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, F_{SM}, F_{SR}) = \mathbf{J}_{TS}^T \begin{bmatrix} F_{SM} - F_{SR} \\ 0 \end{bmatrix} - \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} - \frac{\partial R(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} , \tag{67}$$

with the drive force of the carriage $F_{SM}$ and the associated friction force $F_{SR}$.

Here, the potential energy $U(\mathbf{q})$ consists of the gravity potential of all rigid and elastic bodies, as well as the strain energy of the elastic beam. The Rayleigh function $R(\dot{\mathbf{q}})$ allows for an efficient computation of the stiffness-proportional damping matrix.

After a linearization for small bending deflections, the equations of motion can be stated in M-D-K form

$$\mathbf{M}\ddot{\mathbf{q}}\left(t\right)+\mathbf{D}\dot{\mathbf{q}}\left(t\right)+\mathbf{K}\mathbf{q}\left(t\right)=\mathbf{h}\cdot\left(F_{SM}\left(t\right)-F_{SR}\left(\dot{y}_{S}\left(t\right)\right)\right)\ . \tag{68}$$

The symmetric mass matrix is given by

$$\mathbf{M}(\kappa)=\left[\begin{array}{ccc} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{array}\right] \tag{69}$$

with

$$m_{11}=m_{S}+\rho Al+m_{K}+m_{E}\ , \tag{70}$$

$$m_{12}=\frac{3}{8}\rho Al+\frac{m_{K}\kappa^{2}}{2}\left(3-\kappa\right)+m_{E}\ , \tag{71}$$

$$m_{13}=\frac{1}{3}\rho Al+m_{K}\cdot\kappa^{2}+m_{E}\ , \tag{72}$$

$$m_{22}=\frac{33}{140}\rho Al+\frac{6\rho I_{zB}}{5l}+\frac{m_{K}\kappa^{4}}{4}\left(3-\kappa\right)^{2}+\frac{9\theta_{K}\kappa^{2}}{4l^{2}}\left(2-\kappa\right)^{2}+m_{E}\ , \tag{73}$$

$$m_{23}=\frac{13}{60}\rho Al+\frac{5\rho I_{zB}}{4l}+\frac{m_{K}\kappa^{4}}{2}\left(3-\kappa\right)+\frac{3\theta_{K}\kappa^{2}}{l^{2}}\left(2-\kappa\right)+m_{E}\ , \tag{74}$$

$$m_{33}=\frac{1}{5}\rho Al+\frac{4\rho I_{zB}}{3l}+m_{K}\kappa^{4}+\frac{4\theta_{K}\kappa^{2}}{l^{2}}+m_{E}\ . \tag{75}$$

The damping matrix, which is assumed to be stiffness-proportional, and the stiffness matrix become

$$\mathbf{D}=\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & \frac{3k_{d}EI_{zB}}{l^{3}} & \frac{3k_{d}EI_{zB}}{l^{3}} \\ 0 & \frac{3k_{d}EI_{zB}}{l^{3}} & \frac{4k_{d}EI_{zB}}{l^{3}} \end{array}\right]\ ,\quad \mathbf{K}(\kappa)=\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & k_{22} & k_{23} \\ 0 & k_{23} & k_{33} \end{array}\right]\ , \tag{76}$$

with

$$k_{22}=\frac{3EI_{zB}}{l^{3}}-\frac{3}{8}\rho Ag-\frac{3m_{K}g\kappa^{3}}{l}\left(1+\frac{3\kappa^{2}}{20}-\frac{3\kappa}{4}\right)-\frac{6m_{E}g}{5l}\ , \tag{77}$$

$$k_{23}=\frac{3EI_{zB}}{l^{3}}-\frac{7}{20}\rho Ag+\frac{m_{K}g\kappa^{3}}{l}\left(\frac{3\kappa}{4}-2\right)-\frac{5m_{E}g}{4l}\ , \tag{78}$$

$$k_{33}=\frac{4EI_{zB}}{l^{3}}-\frac{1}{3}\rho Ag-\frac{4m_{K}g\kappa^{3}}{3l}-\frac{4m_{E}g}{3l}\ . \tag{79}$$

In (76), the parameter $k_d$ denotes the coefficient of stiffness-proportional damping for the elastic beam. The input vector of the generalized forces, which accounts for the control input $F_{SM}$ as well as the disturbance input $F_{SR}$, reads

$$\mathbf{h}=\left[\begin{array}{ccc} 1 & 0 & 0 \end{array}\right]^{T}\ . \tag{80}$$

The electric drive for the carriage is operated with a fast underlying velocity control on its current converter. The resulting dynamic behavior is characterized by a first-order lag system with a time constant $T_{1y}$

$$T_{1y}\ddot{y}_{S}\left(t\right)+\dot{y}_{S}\left(t\right)=v_{S}\left(t\right)-v_{S0}\ , \tag{81}$$

where the input disturbance $v_{S0}$ represents remaining uncertainties. In the following, this differential equation replaces the equation of motion for the carriage in the

mechanical system model (68), which leads to a modified mass matrix as well as a modified damping matrix

$$\mathbf{M}_y(\kappa) = \begin{bmatrix} T_{1y} & 0 & 0 \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \quad , \quad \mathbf{D}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3k_d EI_{zB}}{l^3} & \frac{3k_d EI_{zB}}{l^3} \\ 0 & \frac{3k_d EI_{zB}}{l^3} & \frac{4k_d EI_{zB}}{l^3} \end{bmatrix} \quad . \tag{82}$$

The stiffness matrix $\mathbf{K}_y(\kappa) = \mathbf{K}(\kappa)$ and the input vector for the generalized forces $\mathbf{h}_y = \mathbf{h}$, however, remain unchanged. Hence, the equations of motion are given by

$$\ddot{\mathbf{q}} = -\mathbf{M}_y^{-1}\mathbf{K}_y\mathbf{q} - \mathbf{M}_y^{-1}\mathbf{D}_y\dot{\mathbf{q}} + \mathbf{M}_y^{-1}\mathbf{h}_y v_S - \mathbf{M}_y^{-1}\mathbf{h}_y v_{S0} \ , \tag{83}$$

with the carriage velocity $v_S$ as the new control input $u_y$.

For feedforward and feedback control design, a vanishing input disturbance $v_{S0}$ is considered, and the system representation is reformulated in the state-space form

$$\dot{\mathbf{x}}_y = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_y^{-1}\mathbf{K}_y & -\mathbf{M}_y^{-1}\mathbf{D}_y \end{bmatrix}}_{\mathbf{A}_y(\kappa)} \underbrace{\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}}_{\mathbf{x}_y} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}_y^{-1}\mathbf{h}_y \end{bmatrix}}_{\mathbf{b}_y(\kappa)} \underbrace{v_S}_{u_y} \ . \tag{84}$$

In the remainder of this paper, it is assumed that the cage is positioned at a fixed, exactly known, position $\kappa$. First extensions of the simulation approach to a joint control of the carriage and the cage can be found in [28].

For the following simulation, the control law

$$u_y = -\mathbf{k}_y^T \cdot \mathbf{x}_y \tag{85}$$

is designed in such a way that the constant operating point $\mathbf{x}_y = \mathbf{0}$ is stabilized. For that purpose, the controller gain $\mathbf{k}_y$ is determined by a minimization of the quadratic cost function

$$J = \frac{1}{2}\int_0^\infty \left(\mathbf{x}_y^T\mathbf{Q}_y\mathbf{x}_y + r_y u_y^2\right)\mathrm{d}t \ , \tag{86}$$

which contains the positive definite weighting matrix $\mathbf{Q}_y$ for the penalization of state errors and the scalar weight $r_y$ for a quantification of the control effort. In the following, the weighting matrix $\mathbf{Q}_y$ is chosen as the diagonal matrix

$$\mathbf{Q}_y = \mathrm{diag}\left[7500\,, 5000\,, 5000\,, 0.01\,, 0.01\,, 0.01\right] = \mathrm{const} > 0 \ , \tag{87}$$

with the input weighting factor $r_y = 10 = \mathrm{const}$.

Considering the system parameters listed in [1], these settings lead to the following eigenvalues of the closed-loop system:

$$\begin{aligned} \lambda_1 &= -96.2212226512099 + 334.8463405769027j \\ \lambda_2 &= -96.2212226512099 - 334.8463405769027j \\ \lambda_3 &= -118.2015893613681 \\ \lambda_4 &= -32.7889993139425 \\ \lambda_5 &= -13.6916015502190 + 34.7570591820436j \\ \lambda_6 &= -13.6916015502190 - 34.7570591820436j \end{aligned} \tag{88}$$

Despite the complexity of this system model and the quite large initial state uncertainties

$$\mathbf{x}_y(0) \in \begin{bmatrix} 0.2000 \cdot [0.95\ ;\ 1.05] \\ 0.0010 \cdot [0.95\ ;\ 1.05] \\ 0.0005 \cdot [0.95\ ;\ 1.05] \\ [0\ ;\ 0] \\ [0\ ;\ 0] \\ [0\ ;\ 0] \end{bmatrix}, \tag{89}$$

ValEncIA-IVP manages to simulate this system model with good accuracy. The resulting enclosures for two selected state variables determined by ValEncIA-IVP and VNODE-LP are displayed in Figs. 10 and 11. Note again that the increased interval diameters in the initial conditions of the exponential enclosure approach result from a forward and backward transformation of a real-valued initial interval box into the complex plane with an unavoidable wrapping effect.

Future work will deal with the use of this simulation routine for the validation of tracking control strategies in which the load mass is no longer located at a fixed position $\kappa = 0.5$. Instead, two different scenarios will be studied:

1. An accurate underlying controller is available for the load position. Hence, $\kappa$ can be used as a time-varying parameter during the simulation.
2. A robust controller will be designed which only relies on worst-case bounds for $\kappa$. Hence, this value has to be treated as an interval parameter. This is a challenging task since $\kappa$ enters the state equations in terms of rational expressions, in which its maximum power is $\kappa^{10}$.

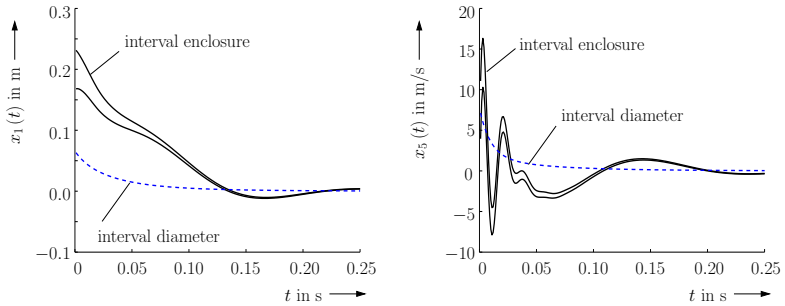


Figure 10: State enclosures computed by the complex-valued exponential enclosure approach in ValEncIA-IVP for the controlled rack feeder system.

# 6 Conclusions and Outlook on Future Work

In this paper, an efficient exponential enclosure approach has been presented for dynamic system models with an oscillatory behavior. It is based on complex-valued interval arithmetic and serves as an extension of the basic enclosure techniques that are available in ValEncIA-IVP.

Future work will deal with the development of techniques for the reduction of overestimation in complex-valued expressions which can be used as generalizations of the

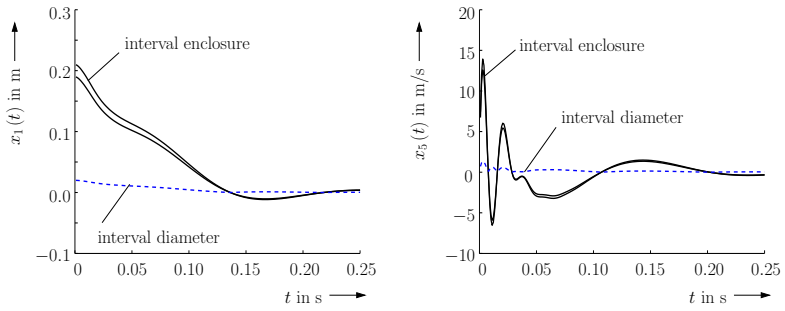


Figure 11: State enclosures computed by VNODE-LP for the controlled rack feeder system.

monotonicity tests that are employed in the real-valued implementation of VALENCIA-IVP. In such a way, it will become possible to use the presented techniques for further nonlinear system models, as well as to extend it to the solution of IVPs for differential-algebraic equations in VALENCIA-IVP [25].

Finally, the presented algorithm will be used within a framework for an interval-based predictive control design for linear and nonlinear uncertain systems in which verified state enclosures have to be computed in real time. This goal can be reached by the presented techniques since, for example, the simulation of the rack feeder system already can be carried out in real time, although only a prototypical implementation has been used in INTLAB. Further research work will, however, also focus on a C++ implementation in C-XSC [11], which could be shown by the authors to be compatible with real-time systems for rapid control prototyping, cf. [6].

# A Details on the Derivation of the Basic Iteration Scheme in VALENCIA-IVP

Assume that there exists a converging iteration procedure

$$[\mathbf{x}]^{(\kappa+1)}(t) \subseteq [\mathbf{x}]^{(\kappa)}(t) \quad \text{for all} \quad \kappa \geq 0 \tag{90}$$

that is characterized by the following two properties:

1. $[\mathbf{x}]^{(\kappa)}(t)$ and $[\mathbf{x}]^{(\kappa+1)}(t)$ contain all possible true solutions $\mathbf{x}^*(t)$ of the IVP

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \ , \quad \mathbf{x}(0) = \mathbf{x}^*(0) = \mathbf{x}_0 \tag{91}$$

   with certainty, and

2. $[\mathbf{x}]^{(\kappa)}([0\ ;\ T])$ and $[\mathbf{x}]^{(\kappa+1)}([0\ ;\ T])$, $T > 0$, are inclusion monotonic bounds for the range of $\mathbf{x}^*(t)$ over the complete time interval $[0\ ;\ T] \ni t$.

Using the above-stated properties,

$$\mathbf{f}\left([\mathbf{x}]^{(\kappa)}([0\ ;\ T])\right) \quad \text{and} \quad \mathbf{f}\left([\mathbf{x}]^{(\kappa+1)}([0\ ;\ T])\right) \tag{92}$$

are guaranteed bounds for the exact derivative $\dot{\mathbf{x}}^*(t)$ over the interval $[0\ ;\ T] \ni t$.

Now, an enclosure for the derivative $\dot{\mathbf{x}}^*(t)$ over the interval $[0\ ;\ T]$ is defined according to

$$\dot{\mathbf{x}}^*(t) \in \dot{\tilde{\mathbf{x}}}(t) + [\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T]) \quad , \tag{93}$$

where $\dot{\tilde{\mathbf{x}}}(t)$ is a continuous approximation of $\dot{\mathbf{x}}^*(t)$ for all $t \in [0\ ;\ T]$. This approximation is computed by exact differentiation of a function $\tilde{\mathbf{x}}(t)$ that is assumed to be at least once continuously differentiable for all $t \in [0\ ;\ T]$. Guaranteed enclosures for the range of $\mathbf{x}^*(t)$ over the complete interval $[0\ ;\ \bar{t}\,]$, $\bar{t} \leq T$ are obviously obtained by

$$\begin{aligned} \mathbf{x}^*(t) &\in [\tilde{\mathbf{x}}]([0\ ;\ \bar{t}\,]) + \int_0^{\bar{t}} [\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T])\,\mathrm{d}t \\ &\subseteq [\tilde{\mathbf{x}}]([0\ ;\ \bar{t}\,]) + [\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T]) \cdot [0\ ;\ \bar{t}\,] \\ &=: [\tilde{\mathbf{x}}]([0\ ;\ \bar{t}\,]) + [\mathbf{R}]^{(\kappa)}([0\ ;\ \bar{t}\,]) =: [\mathbf{x}]^{(\kappa)}([0\ ;\ \bar{t}\,]) \end{aligned} \tag{94}$$

with time-invariant bounds of $[\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T])$. In (94), $[\tilde{\mathbf{x}}]([0\ ;\ \bar{t}\,])$ denotes the range of the approximate solution over the corresponding time interval.

Moreover, the solution $\mathbf{x}^*(\bar{t}\,)$ at the point $t = \bar{t}$ can be enclosed by

$$\mathbf{x}^*(\bar{t}\,) \in [\mathbf{x}]^{(\kappa)}(\bar{t}\,) = \tilde{\mathbf{x}}(\bar{t}\,) + [\dot{\mathbf{R}}]^{(\kappa)}([0\ ;\ T]) \cdot \bar{t} \quad . \tag{95}$$

This justifies the applicability of (8)–(16) and can be transferred in a straightforward manner to the reasoning for the exponential enclosure technique described in Sec. 3.2.

# B Details on the Derivation of the Exponential Iteration Scheme in VALENCIA-IVP

A converging iteration process for the exponential enclosure technique in VALENCIA-IVP is characterized by

$$\exp\left([\mathbf{\Lambda}]^{(\kappa+1)} \cdot t\right) \cdot [\mathbf{x}_e](0) \subseteq \exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot t\right) \cdot [\mathbf{x}_e](0) \quad . \tag{96}$$

Using a componentwise notation, this expression is equivalent to

$$\exp\left([\lambda_i]^{(\kappa+1)} \cdot t\right) \cdot [x_{e,i}](0) \subseteq \exp\left([\lambda_i]^{(\kappa)} \cdot t\right) \cdot [x_{e,i}](0) \ , \quad i = 1, \ldots, n \tag{97}$$

with

$$\begin{aligned} &\left[\exp\left(\underline{\lambda}_i^{(\kappa+1)} \cdot t\right)\ ;\ \exp\left(\overline{\lambda}_i^{(\kappa+1)} \cdot t\right)\right] \cdot [x_{e,i}](0) \\ &\quad \subseteq \left[\exp\left(\underline{\lambda}_i^{(\kappa)} \cdot t\right)\ ;\ \exp\left(\overline{\lambda}_i^{(\kappa)} \cdot t\right)\right] \cdot [x_{e,i}](0) \quad . \end{aligned} \tag{98}$$

Because the exponential state enclosure technique is only defined for initial state intervals $[\mathbf{x}_e](0)$ that do not contain the value zero in any of their components $[x_{e,i}](0)$ (cf. (17)), the enclosure property $[\lambda_i]^{(\kappa+1)} \subseteq [\lambda_i]^{(\kappa)}$ can be checked in the following way. The expression

$$\begin{aligned} &\left[\exp\left(\underline{\lambda}_i^{(\kappa+1)} \cdot t\right) \cdot \underline{x}_{e,i}(0)\ ;\ \exp\left(\overline{\lambda}_i^{(\kappa+1)} \cdot t\right) \cdot \overline{x}_{e,i}(0)\right] \\ &\quad \subseteq \left[\exp\left(\underline{\lambda}_i^{(\kappa)} \cdot t\right) \cdot \underline{x}_{e,i}(0)\ ;\ \exp\left(\overline{\lambda}_i^{(\kappa)} \cdot t\right) \cdot \overline{x}_{e,i}(0)\right] \end{aligned} \tag{99}$$

directly follows from (98). For $\underline{x}_{e,i}(0) > 0$, the relation (99) implies

$$\exp\left(\underline{\lambda}_i^{(\kappa+1)} \cdot t\right) \geq \exp\left(\underline{\lambda}_i^{(\kappa)} \cdot t\right) \quad \text{and} \quad \exp\left(\overline{\lambda}_i^{(\kappa+1)} \cdot t\right) \leq \exp\left(\overline{\lambda}_i^{(\kappa)} \cdot t\right) \tag{100}$$

for $t \geq 0$. Due to the monotonicity of the exponential function, (100) can be reformulated as

$$\underline{\lambda}_i^{(\kappa+1)} \geq \underline{\lambda}_i^{(\kappa)} \quad \text{and} \quad \overline{\lambda}_i^{(\kappa+1)} \leq \overline{\lambda}_i^{(\kappa)} \ , \tag{101}$$

which is equivalent to the equation (27). This proof can be performed in an analogous way for $\overline{x}_{e,i}(0) < 0$.

Validity of (29) can, furthermore, be shown by

$$\begin{aligned} [\lambda_i]^{(\kappa+1)} &:= \frac{f_i\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right)}{\exp\left([\lambda_i]^{(\kappa+1)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)} \\ &\subseteq \left[\tilde{\lambda}_i\right]^{(\kappa+1)} := \frac{f_i\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right)}{\exp\left([\lambda_i]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)} \ , \end{aligned} \tag{102}$$

which holds due to the fact that

$$\exp\left([\lambda_i]^{(\kappa+1)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0) \subseteq \exp\left([\lambda_i]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0) \tag{103}$$

corresponds to

$$\frac{1}{\exp\left([\lambda_i]^{(\kappa+1)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)} \subseteq \frac{1}{\exp\left([\lambda_i]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)} \ . \tag{104}$$

A conservative overapproximation of (102) is then given by the redefined coefficient interval

$$[\lambda_i]^{(\kappa+1)} := \frac{f_i\left(\exp\left([\mathbf{\Lambda}]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [\mathbf{x}_e]\,(0)\right)}{\exp\left([\lambda_i]^{(\kappa)} \cdot [0\ ;\ T]\right) \cdot [x_{e,i}]\,(0)} \ . \tag{105}$$

# References


[1] H. Aschemann, D. Schindele, and J. Ritzke. State and Disturbance Estimation for Robust Control of Fast Flexible Rack Feeders. In A. Rauh and E. Auer, editors, *Modeling, Design, and Simulation of Systems with Uncertainties*, Mathematical Engineering, pages 333–351. Springer, Berlin, Heidenberg, 2011.

[2] E. Auer, S. Kiel, and A. Rauh. A Verified Method for Solving Piecewise Smooth Initial Value Problems. *International Journal of Applied Mathematics and Computer Science AMCS*, 23(4), 2013. In print.

[3] E. Auer, A. Rauh, E.P. Hofer, and W. Luther. Validated Modeling of Mechanical Systems with SMARTMOBILE: Improvement of Performance by VALENCIA-IVP. In *Proc. of Dagstuhl Seminar 06021: Reliable Implementation of Real Number Algorithms: Theory and Practice*, Lecture Notes in Computer Science, pages 1–27, 2008.

[4] M. Berz and K. Makino. Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models. *Reliable Computing*, 4:361–369, 1998.

[5] M. Berz and K. Makino. COSY INFINITY Version 8.1. User's Guide and Reference Manual. Technical Report MSU HEP 20704, Michigan State University, 2002.

[6] T. Dötschel, A. Rauh, L. Senkel, and H. Aschemann. Experimental Validation of Interval-Based Sliding Mode Control for Solid Oxide Fuel Cell Systems. In *Proc. of the European Control Conference ECC 2013*, Zurich, Switzerland, 2013.

[7] I. Eble. *Über Taylor-Modelle*. PhD thesis, Universität Karlsruhe (TH), Fakultät für Mathematik, 2007. In German.

[8] T.E. Fortmann and K.L. Hitz. *An Introduction to Linear Control Systems*. Marcel Dekker, Inc., New York, 1977.

[9] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer, Berlin Heidelberg, 2nd edition, 2000.

[10] C. Jordan. *Traité des substitutions et des équations algébriques*. Gauthier-Villars, Paris, 1870. In French.

[11] W. Krämer. XSC Languages (C-XSC, PASCAL-XSC) — Scientific Computing with Validation, Arithmetic Requirements, Hardware Solution and Language Support, n.a. `www.math.uni-wuppertal.de/~xsc/`.

[12] W. Kühn. Rigorous Error Bounds for the Initial Value Problem Based on Defect Estimation. Technical report, 1999. Available online: `http://www.decatur.de/personal/papers/defect.zip`.

[13] Y. Lin and M.A. Stadtherr. Validated solution of initial value problems for ODEs with interval parameters. In *NSF Workshop Proceeding on Reliable Engineering Computing*, Savannah GA, February 22-24 2006.

[14] R. Lohner. Enclosing the Solutions of Ordinary Initial and Boundary Value Problems. In E.W. Kaucher, U.W. Kulisch, and C. Ullrich, editors, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286, Stuttgart, 1987. Wiley-Teubner Series in Computer Science.

[15] R. Lohner. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. PhD thesis, Universität Karlsruhe (TH), Fakultät für Mathematik, 1988. In German.

[16] K. Makino and M. Berz. Suppression of the Wrapping Effect by Taylor Model-Based Validated Integrators. Technical Report MSU HEP 40910, Michigan State University, 2004.

[17] R.E. Moore. *Interval Arithmetic*. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[18] N.S. Nedialkov. *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1999.

[19] N.S. Nedialkov. Interval Tools for ODEs and DAEs. In *CD-Proc. of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006*, Duisburg, Germany, 2007. IEEE Computer Society.

[20] N.S. Nedialkov. Implementing a Rigorous ODE Solver through Literate Programming. In A. Rauh and E. Auer, editors, *Modeling, Design, and Simulation of Systems with Uncertainties*, Mathematical Engineering, pages 3–19. Springer, Berlin, Heidenberg, 2011.

[21] M.S. Petković and L.D. Petković. *Complex Interval Arithmetic and Its Applications*. Wiley-VCH Verlag GmbH, Berlin, 1998.

[22] A. Rauh and E. Auer. Verified Simulation of ODEs and DAEs in VALENCIA-IVP. *Reliable Computing, Special Issue on 13th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2008, El Paso, Texas*, 15(4):370–381, 2011.

[23] A. Rauh, E. Auer, and E.P. Hofer. VALENCIA-IVP: A Comparison with Other Initial Value Problem Solvers. In *CD-Proc. of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006*, Duisburg, Germany, 2007. IEEE Computer Society.

[24] A. Rauh, E. Auer, J. Minisini, and E.P. Hofer. Extensions of VALENCIA-IVP for Reduction of Overestimation, for Simulation of Differential Algebraic Systems, and for Dynamical Optimization. In *PAMM: Proc. of the 6th International Congress on Industrial and Applied Mathematics, Minisymposium Taylor Model Methods and Interval Methods – Applications*, volume 7, pages 1023009–1023010, Zurich, Switzerland, 2007.

[25] A. Rauh, M. Brill, and C. Günther. A Novel Interval Arithmetic Approach for Solving Differential-Algebraic Equations with VALENCIA-IVP. *Special Issue of the International Journal of Applied Mathematics and Computer Science AMCS, "Verified Methods: Applications in Medicine and Engineering"*, 19(3):381–397, 2009.

[26] A. Rauh, J. Kersten, E. Auer, and H. Aschemann. Sensitivity Analysis for Reliable Feedforward and Feedback Control of Dynamical Systems with Uncertainties. In *Proc. of 8th Intl. Conference on Structural Dynamics EURODYN 2011*, Leuven, Belgium, 2011.

[27] A. Rauh, J. Kersten, E. Auer, and H. Aschemann. Sensitivity-Based Feedforward and Feedback Control for Uncertain Systems. *Computing*, (2–4):357–367, 2012.

[28] A. Rauh, R. Westphal, and H. Aschemann. Verified Simulation of Control Systems with Interval Parameters Using an Exponential State Enclosure Technique. In *CD-Proc. of IEEE Intl. Conference on Methods and Models in Automation and Robotics MMAR*, Miedzyzdroje, Poland, 2013.

[29] S.M. Rump. INTLAB — INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluver Academic Publishers, 1999.

[30] A.A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, 2005.